



Tecnológico de Monterrey

Guía de entrega de plan de calidad y bitácora de calidad

Manuel Eduardo Covarrubias Rodríguez A01737781

Javier Cuatepotzo Hernandez A01737789

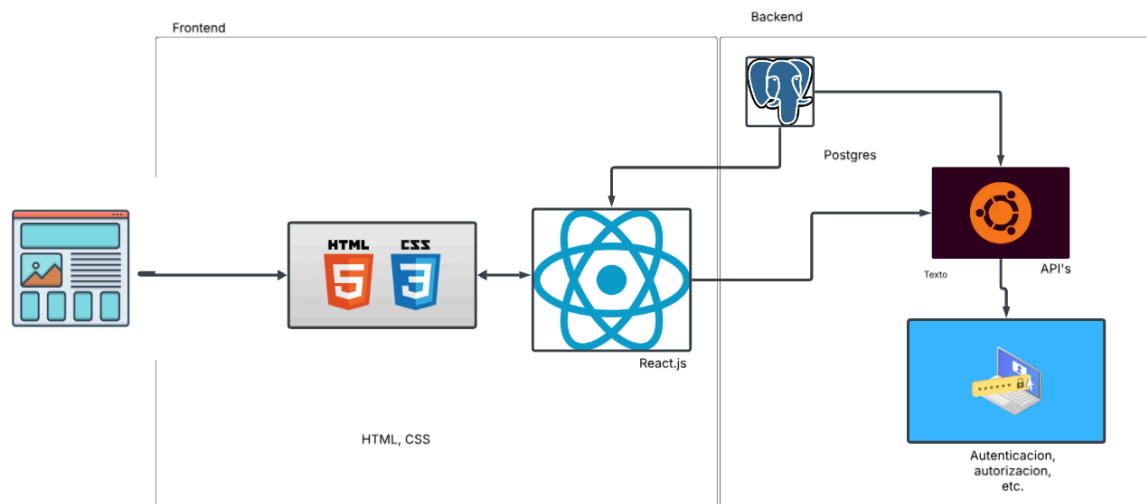
Anhuar Maldonado Aguilar A01737767

Juan Daniel Salmerón Mora A01737637

6 de junio del 2025

1. Arquitectura de Software

+ Diagrama de componentes.



- Roles de cada componente

Este diagrama muestra un flujo en el que el usuario interactúa con el frontend (React ++ HTML/CSS), que a su vez se comunica con el backend (API sobre Ubuntu), el cual gestiona datos en PostgreSQL y servicios de autenticación/autorización.

Este diagrama de componentes representa la arquitectura de una aplicación web dividida en frontend y backend:

Frontend

- **HTML y CSS:** Se usan para estructurar y dar estilo básico a la interfaz de usuario.
- **React.js:** Biblioteca de JavaScript utilizada para construir la interfaz de usuario de manera dinámica y modular.

Backend

- **React.js.** también actúa como intermediario con el backend mediante llamadas a API.
- **Ubuntu:** Servidor donde se alojan las API's que gestionan la lógica del servidor y acceso a datos.
- **PostgreSQL:** Base de datos que almacena la información.
- **Autenticación y autorización:** Sistema encargado de validar usuarios, gestionar accesos y seguridad.

- Flujos principales de interacción:

Registros:

Navegador solicita URL → se entrega HTML/CSS → se envían datos de los formularios → React y los endpoints se encargan de enviar la información correctamente a la base de datos.

Login:

React envía credenciales a /auth/login → Backend → Auth valida con Postgres → devuelve token → React lo almacena y redirige.

Lectura/Actualización de datos protegidos:

React solamente deja ingresar mostrar PII información a las coordinadoras de sede

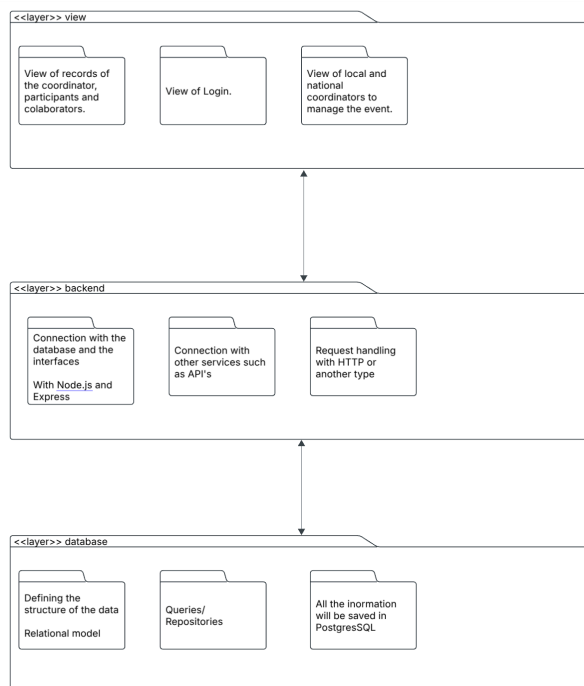
- *Cambios relevantes respecto a versiones anteriores.*

Al principio del desarrollo de este diagrama se esperaba usar la nube para poder hacer de host de la página, sin embargo decidimos no hacerlo y probamos mejor un servidor local en ubuntu. Por otra parte, en el diseño decidimos hacer todo con CSS básico sin usar alguna otra librería externa.

- *Link:*

[Link de diagrama de componentes](#)

+ *Diagrama de paquetes o módulos.*



- *Roles de cada componente*

Este diagrama de paquetes muestra una arquitectura en tres capas:

- **Vista (View):** Interfaces para login, registros y gestión de coordinadores.
- **Backend:** Lógica con Node.js y Express, conexión a APIs y manejo de peticiones.
- **Base de datos:** Estructura y consultas en PostgreSQL.

- *Flujos principales de interacción:*

Se dispone de varias vistas en el frontend (login; registro y listado de sedes, participantes y mentoras; panel de cada coordinadora). Cuando el usuario interactúa con cualquiera de estas vistas, React envía peticiones HTTP (GET, POST, PUT, DELETE) a los endpoints del backend (Node.js/Express). El backend recibe cada solicitud, valida permisos y datos, ejecuta las consultas o modificaciones correspondientes en PostgreSQL, y devuelve la respuesta en formato JSON.

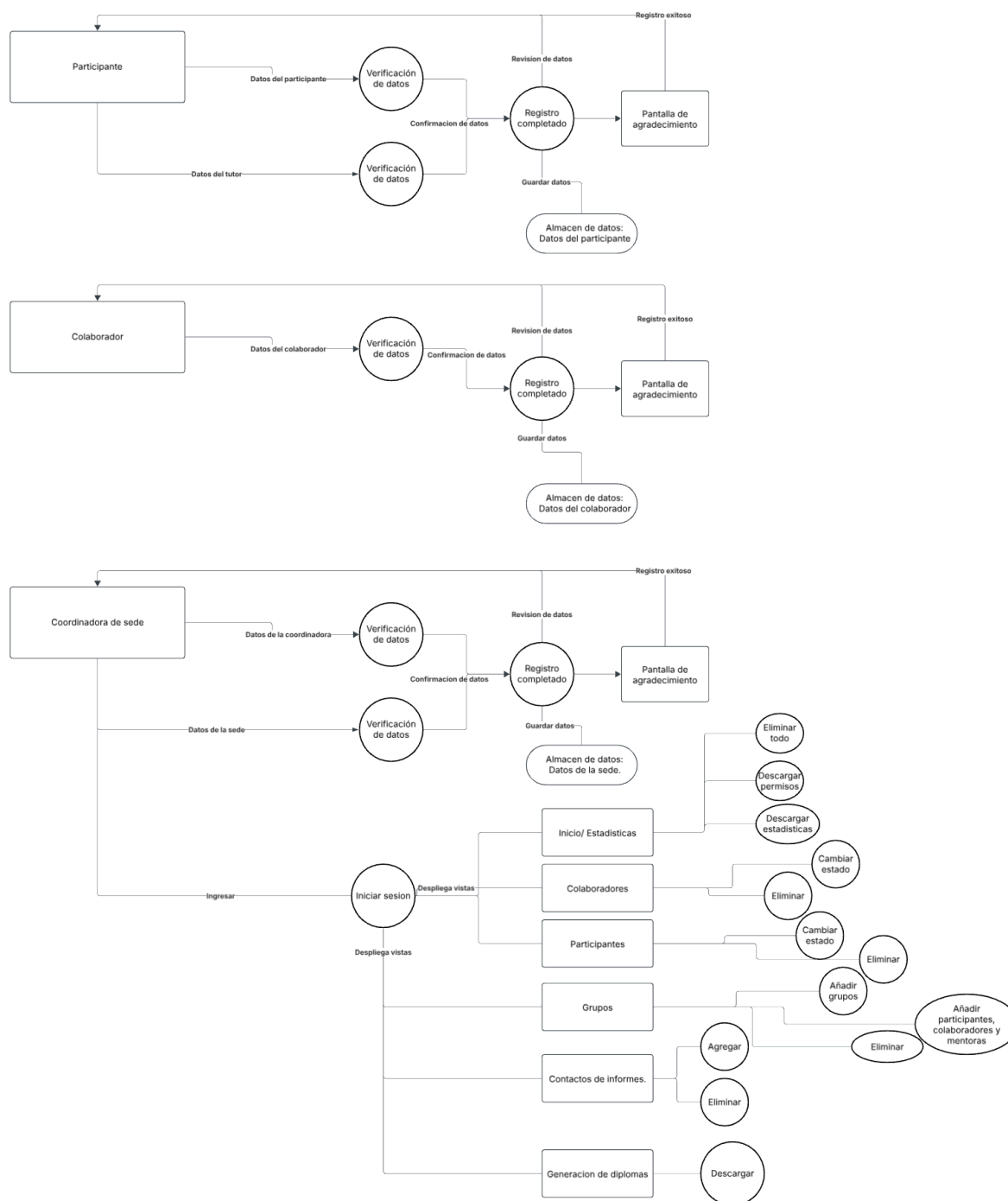
- *Cambios relevantes respecto a versiones anteriores.*

No hay cambios relevantes respecto a versiones anteriores; el diagrama se ha mantenido íntegro desde el inicio, siguiendo los requerimientos de la socio formadora.

- *Link:*

[Link diagrama de paquetes](#)

+ Diagrama de flujo de datos.



- Roles de cada componente

El diagrama presenta la estructura funcional de una plataforma para el registro y gestión de participantes, colaboradores y coordinadores de sede. Cada tipo de usuario tiene un rol específico: los participantes ingresan sus datos, que son verificados y almacenados, concluyendo el proceso con una pantalla de agradecimiento. Los colaboradores siguen un flujo similar, mientras que los coordinadores de sede tienen además acceso a funciones administrativas como visualizar estadísticas, gestionar usuarios y generar diplomas.

- Flujos principales de interacción:

Los flujos principales de interacción comienzan con el ingreso de datos, seguidos por su verificación y almacenamiento. En el caso del coordinador, se incluye un proceso adicional de inicio de sesión para acceder a paneles que permiten agregar, modificar, eliminar y descargar información relacionada con grupos, participantes y colaboradores.

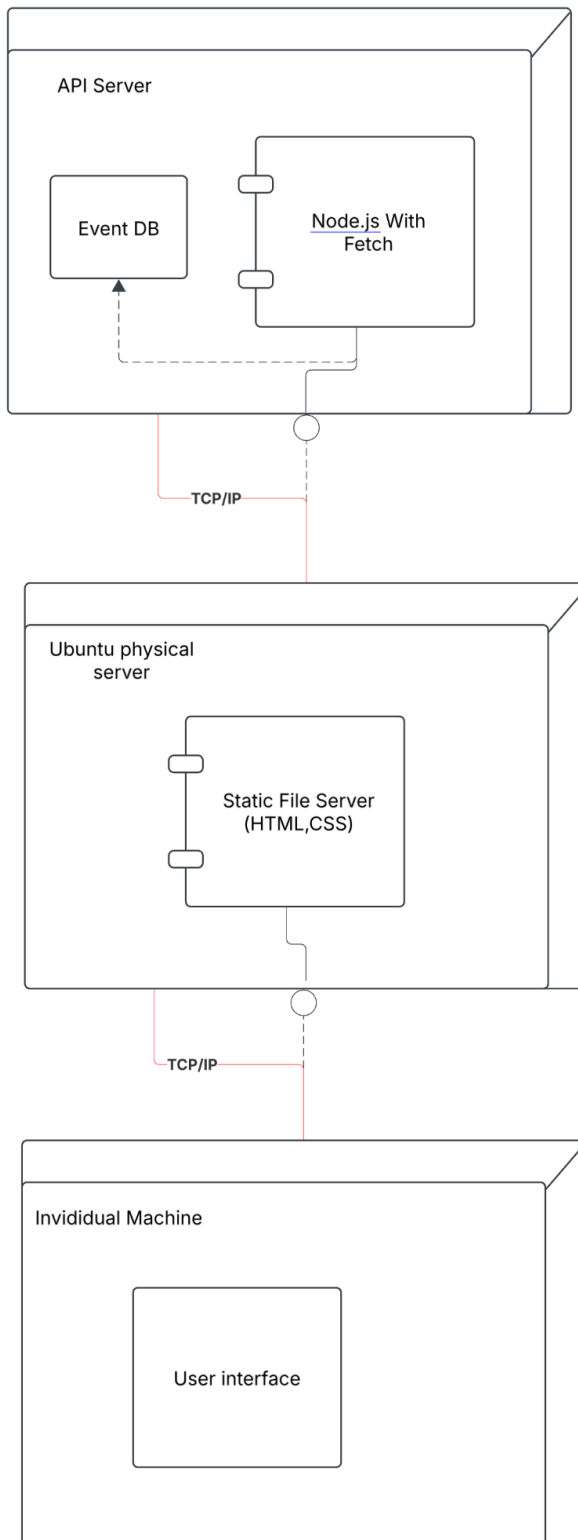
- *Cambios relevantes respecto a versiones anteriores.*

En comparación con versiones anteriores, esta nueva versión introduce una interfaz más robusta para los coordinadores, permitiendo una gestión más completa del sistema. Se destacan nuevas funcionalidades como agregar registros de personas de contacto, agregar coordinadoras asociadas que eso no tiene el flujo, pero que esta implementado por petición del SF.

- *Link:*

[Diagrama flujo de datos](#)

+ *Diagrama de despliegue (si aplica).*



- *Roles de cada componente*

Este es un diagrama de despliegue que muestra cómo se distribuyen los componentes del sistema en distintos entornos físicos:

- **Máquina del usuario:** Contiene la interfaz de usuario que accede a la aplicación.
- **Servidor físico Ubuntu:** Sirve archivos estáticos (HTML, CSS) desde un servidor web
- **Servidor API:** Usa Node.js para manejar peticiones y acceder a la base de datos de eventos.

- *Flujos principales de interacción:*

Las comunicaciones entre componentes se hacen a través de **protocolo TCP/IP**, conectando al usuario con el servidor web y este con el servidor de API.

- *Cambios relevantes respecto a versiones anteriores.*

Aquí también tenemos el cambio del servidor ubuntu, al principio teníamos la idea de desplegar nuestra aplicación en la nube AWS.

- *Link:*

[Link diagrama de despliegue](#)

+ **Esquema de base de datos (modelo lógico actual).**

- **Modelo relacional:**



- **Roles de cada componente**

Cada recuadro representa una tabla (por ejemplo: Participantes, Sede, Mentor, Colaboradores, etc.). Se destacan las llaves primarias (id_...) y las llaves foráneas que enlazan entidades (como id_sede, id_tutor, id_grupo).

Las tablas incluyen atributos clave como nombres, correos, niveles, estados, contraseñas y archivos. Refleja las relaciones entre entidades: por ejemplo, un participante está relacionado con una sede, grupo y tutor; una sede tiene una coordinadora y puede tener asociadas.

- **Flujos principales de interacción:**

Las tablas están estrechamente relacionadas en la estructura de nuestra base de datos. Las llaves foráneas como id_sede, id_coordinadora e id_grupo se representan en color verde. Por su parte, las llaves primarias sobre las cuales se basan las operaciones de los endpoints (GET, POST, PUT, UPDATE y DELETE) aparecen en color amarillo.

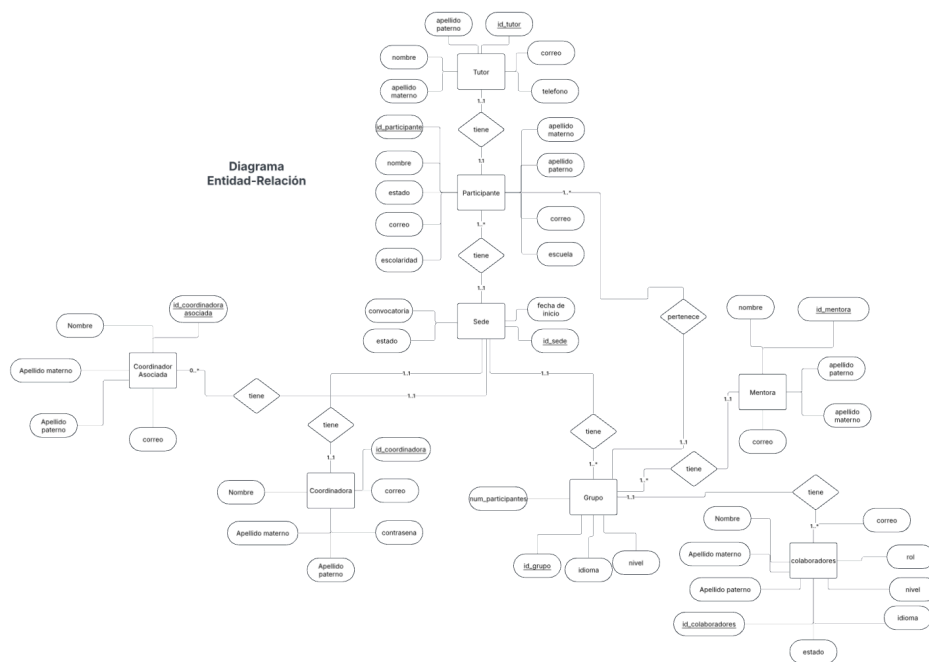
- **Cambios relevantes respecto a versiones anteriores.**

Este es el que más cambios ha sufrido debido a que tuvimos que hacer modificaciones en base a requerimientos del socio-formador o incluso hicimos algunos cambios en base a si lo veíamos viable para poder hacerlo.

- **Link:**

[Link modelo relacional](#)

- Modelo entidad - relación:



- Rol de los componentes:

Participante: Persona registrada que pertenece a una sede y a un grupo. Tiene un tutor asignado.

Tutor: Encargado del participante, con relación 1 a 1.

Sede: Lugar donde se agrupan participantes y se asignan coordinadoras.

Coordinadora / Coordinadora Asociada: Encargadas de la sede. La asociada es opcional.

Grupo: Clasificación de participantes por idioma y nivel, con una mentora asignada.

Mentora: Responsable académica de un grupo.

Colaboradores: Usuarios del sistema con distintos roles de gestión.

- Flujos principales de interacción:

Un participante se registra con su tutor y se asigna a una sede.

La sede tiene una coordinadora (y opcionalmente una asociada).

El participante se agrupa en un grupo con un idioma y nivel, al cual se asigna una mentora.

Colaboradores administran el sistema según su rol.

- Cambios relevantes:

Se agregó la entidad Colaboradores con roles definidos.

Se integró la relación directa entre Mentora y Grupo.

- Link:

[Diagrama entidad - relación](#)

2. Heurísticas de usabilidad.

- Documento con todas las heurísticas del proyecto: [☰ Validación de Software](#)

3. Pruebas funcionales.

- Documentos de pruebas funcionales [✚ Pruebas funcionales](#)

4. Casos de prueba

- Documento de casos de prueba [☰ Casos de prueba](#)

5. Administración de riesgos

- Documento de gestión de riesgos: [W Gestión de riesgos.docx](#)

6. Pruebas de UX

- Documentos de pruebas UX [☰ Evaluación de Usabilidad con Cámara Gesell](#)
[☰ Segundas pruebas](#)

7. Tareas disponibles en el sistema.

Dentro de nuestro sistema se pueden realizar las siguientes tareas:

1. Log in para administradora nacional.
2. Log in para administradora de sede.
3. Registro de participantes.
4. Registro de colaboradora.
5. Registro de sede.
6. Aceptar cada registro.
7. Rechazar cada registro.
8. Editar cada registro.
9. Eliminar cada registro.
10. Registrar contactos para información para participantes/colaboradores/sedes.
11. Registrar coordinadora asociada.
12. Registrar mentoras.
13. Aceptar/rechazar/eliminar registro de mentoras.
14. Aceptar/rechazar/eliminar registro de coordinadora asociada.
15. Visualización de documentos de registros (participantes y sedes)

16. Creación de grupos para cada sede como administradora de sede.
17. Asignar participante/colaboradora a grupo.
18. Eliminar grupo.
19. Descargar documentación de participantes.
20. Creación de reconocimientos para participantes/colaboradores/organizadores.
21. Eliminar base de datos completa.
22. Descargar estadísticas en excel.

Durante el desarrollo tuvimos dos experimentos en la cámara gesell para evaluar nuestro sistema.

Durante la primera prueba únicamente se realizó el testeo de los registros de sedes, colaboradores y participantes, durante estos experimentos se recibieron algunos comentarios para realizar ciertos cambios.

Durante la segunda prueba se evaluaron todas las funcionalidades restantes del sistema. Esto incluyó el inicio de sesión para las administradoras (tanto nacional como de sede), la edición, eliminación, aceptación y rechazo de todos los tipos de registros, así como la asignación de participantes y colaboradoras a grupos. También se puso a prueba la creación y eliminación de grupos, la visualización y descarga de documentos, el registro y gestión de mentoras y coordinadoras asociadas, el registro de contactos informativos, la creación de reconocimientos para los distintos perfiles del sistema, la descarga de estadísticas en formato Excel, y la eliminación completa de la base de datos. Gracias a estos ensayos, se identificaron oportunidades para mejorar la usabilidad del sistema y se validó la funcionalidad general del mismo.